IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Suggested resolution to mesh comments | | | | |
| Date: 2019-03-20 | | | | |
| Author(s): | | | | |
| Name | Company | Address | Phone | Email |
| Kazuyuki Sakoda | Sony |  |  | Kazuyuki.Sakoda (at) sony (dot) com |
|  |  |  |  |  |
|  |  |  |  |  |

Abstract

This document provides suggested resolutions to CIDs 2206, 2331, 2333, 2334, 2335, and 2474. They are all related to mesh STA.

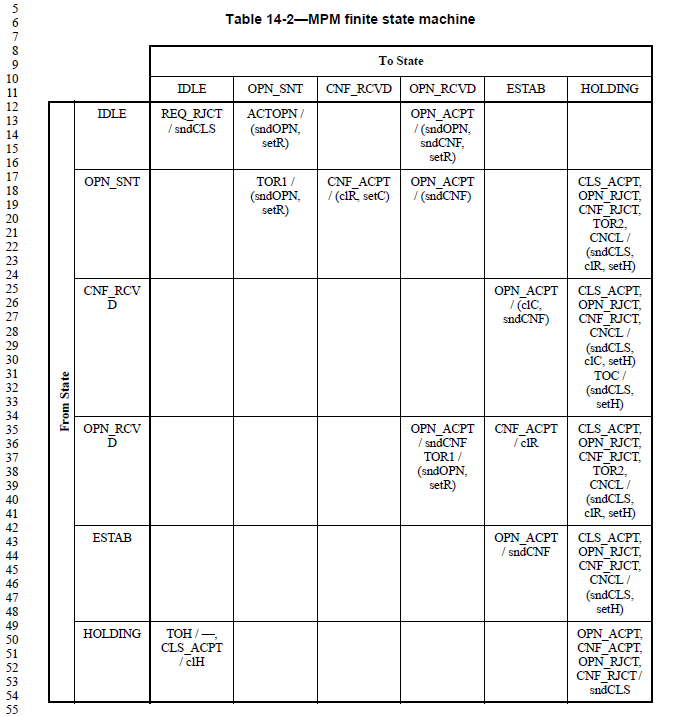
R0: initial proposal.

# CID 2206 Comment:

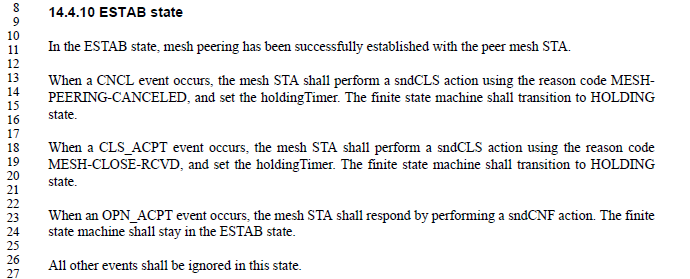
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **PP.LL** | **Comment** | **Proposed Change** | **Suggested Resolution** |
| 2206 | 2752.26 | MPM FSM description of the ESTAB state actions in 14.4.10 is not consistent with the MPM finite state machine definition in Table 14-2: description of the OPN\_RJCT and CNF\_RJCT events is missing. Because of this, the "All other events shall be ignored in this state" requirement would be followed and that would not be consistent with Table 14-2. | On page 2752 line 21, insert two new paragraphs:  "When an OPN\_RJCT event occurs, the mesh STA shall clear the confirmTimer, perform a sndCLS action using the reason code as specified by the OPN\_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state."  and  "When a CNF\_RJCT event occurs, the mesh STA shall clear the confirmTimer, perform a sndCLS action using the reason code as specified by the CNF\_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.". | REVISED:   Adopt changes proposed in 11-19/429. |

# Discussion:

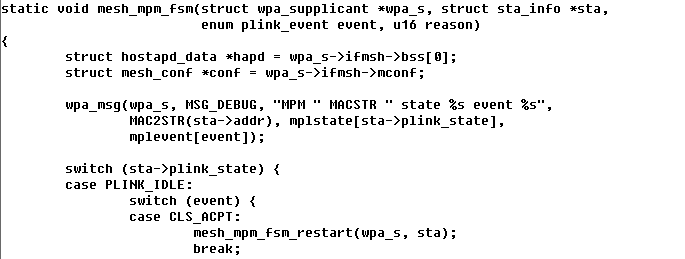
2748.5 reads:



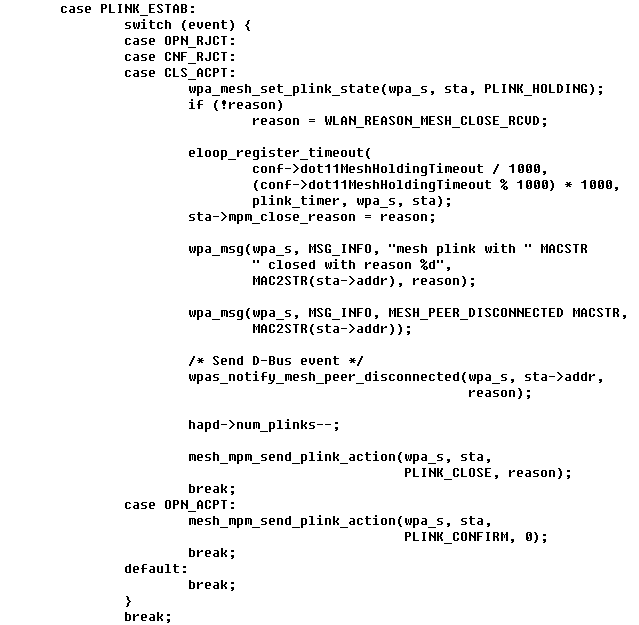
2751.8 reads:



mesh\_mpm.c available at git://w1.fi/hostap.git :



….



The suggested resolution text reads:

When an OPN\_RJCT event occurs, the mesh STA shall clear the confirmTimer, perform a sndCLS action using the reason code as specified by the OPN\_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

The confirmTimer is set per 2750.37, “When a CNF\_ACPT event occurs, the mesh STA shall clear the retryTimer and shall set the confirmTimer and the finite state machine shall transition to CNF\_RCVD state.” It is likely that the confirmTimer is active only when the STA is in CNF\_RCVD state.

It might be more straightforward to remove “clear the confirmTimer” from the suggested text for addition.

# Suggested resolution:

**14.4.10 ESTAB state**

***To REVmd Editor: Change 14.4.10 (ESTAB state) as follows:***

In the ESTAB state, mesh peering has been successfully established with the peer mesh STA.

When a CNCL event occurs, the mesh STA shall perform a sndCLS action using the reason code MESHPEERING-CANCELED, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CLS\_ACPT event occurs, the mesh STA shall perform a sndCLS action using the reason code MESH-CLOSE-RCVD, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When an OPN\_RJCT event occurs, the mesh STA shall perform a sndCLS action using the reason code as specified by the OPN\_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CNF\_RJCT event occurs, the mesh STA shall perform a sndCLS action using the reason code as specified by the CNF\_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When an OPN\_ACPT event occurs, the mesh STA shall respond by performing a sndCNF action. The finite state machine shall stay in the ESTAB state.

All other events shall be ignored in this state.

# CID 2334, 2335 Comment:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **PP.LL** | **Comment** | **Proposed Change** | **Suggested Resolution** |
| 2334 | 2767.32 | The definition of O, "Channel access overhead, which includes frame headers, training  sequences, access protocol frames, etc." is too vague | Change the cited text to "Channel access overhead, which is the time from the start of the first PPDU in the TXOP to the end of the PHY header of the first PPDU in the TXOP that carries includes a QoS Data frame." | REVISED:  Adopt changes proposed in 11-19/429. |
| 2335 | 2768.45 | The definition of O, "Channel access overhead, which includes frame headers, training  sequences, access protocol frames, etc." is too vague | Change the cited text to "Channel access overhead, which is the time from the start of the first PPDU in the TXOP to the end of the PHY header of the first PPDU in the TXOP that carries includes a QoS Data frame." | REVISED:  Adopt changes proposed in 11-19/429. |

# Discussion:

Channel access overhead should represent “MAC overhead” and can be represented more in specific fashion as pointed out by the commenter.

# Suggested resolution:

Revised.

Replace “Channel access overhead, which includes frame headers, training sequences, access protocol frames, etc." with

“

Channel access overhead (in µs), which should include:

* the expected time from the start of the first PPDU in the TXOP to the end of the PHY header of the first PPDU in the TXOP that carries a QoS Data frame that requires immediate acknowledgment, and
* SIFS plus the expected duration of the immediate response to the first PPDU in the TXOP that carries a QoS Data frame

and may include:

* the expected channel access delay, i.e., AIFS plus average random backoff duration

”

# CID 2331 Comment:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **PP.LL** | **Comment** | **Proposed Change** | **Suggested Resolution** |
| 2331 | 2767.32 | There is too much in common between 14.9.2 Airtime link metric and 14.9.3 High PHY rate airtime link metric | Replace everything from the second para of the referenced subclause to the end of it with "This metric is the same as the high PHY rate airtime link metric (see 14.9.3) with <italic>n</italic> set to 1." | REVISED:   Adopt changes proposed in 11-19/429. |

# Discussion:

It is true that there are many common things defined in 14.9.2 (Airtime link metric) and in 14.9.3 (High PHY rate airtime link metric). It might be better to consolidate these subclauses into a subclause.

# Suggested resolution:

***To REVmd Editor: Change 14.9 (Path selection link metric) as follows:***

* Path selection link metric(#109)
* General(#109)

Subclause 14.9 defines a default link metric, the airtime link metric, and another link metric, the high PHY rate airtime link metric, that may be used by a path selection protocol to identify an efficient radio-aware path. A STA should use the high PHY rate airtime link metric when its PHY entity is capable of transmitting frames with PHY rate higher than 54 Mb/s(#2143).

(#109)The extensibility framework allows the default link metric to be overridden by another path selection metric as specified in the mesh profile.







* Airtime link metric and high PHY rate airtime link metric(#109)

The airtime link metric is used when dot11MeshActivePathSelectionMetric is airtimeLinkMetric (1); the high PHY rate airtime link metric is used when dot11MeshActivePathSelectionMetric is highPHYRateAirtimeLinkMetric (2) (see 14.2.3 (Mesh profile)).

Airtime reflects the amount of channel resources consumed by transmitting the frame over a particular link. This measure is approximate and designed for ease of implementation and interoperability.

The airtime for each link is calculated as follows:



where

*O*, *n*, and *Bt* are constants listed in Table 14-6 (Airtime cost constants for high PHY rate airtime link metric(#109))

input parameter *r* is the PHY data rate (in Mb/s). It represents the estimated data rate at which the mesh STA would transmit a (#1556)nominal frame of standard size *Bt* based on current link conditions; its estimation is dependent on local implementation of rate adaptation

input parameter *ef* is the frame error rate for a (#1556)nominal frame of size *Bt*. It is the probability that when a (#1556)nominal frame of standard size *Bt* is transmitted at the PHY data rate *r*, the frame is corrupted due to transmission error; its estimation is a local implementation choice. Failures due to exceeding Mesh TTL should not be included in this estimate as they are not correlated with link performance

The airtime link metric shall be encoded as an unsigned integer in units of 0.01 TU. An example of the airtime link metric is shown in S.5 (Airtime link metric usage example).

The high PHY rate airtime link metric shall be encoded as an unsigned integer in units of 0.01 μs.

|  |  |  |
| --- | --- | --- |
| * Airtime cost constants for airtime link metric and high PHY rate airtime link metric(#109) | | |
| Parameter | Recommended value | Description |
| *O* | Varies depending on PHY | Channel access overhead (in µs), which should include:   * the expected time from the start of the first PPDU in the TXOP to the end of the PHY header of the first PPDU in the TXOP that carries a QoS Data frame that requires immediate acknowledgment, and * SIFS plus the expected duration of the immediate response to the first PPDU in the TXOP that carries a QoS Data frame   and may include:   * the expected channel access delay, i.e., AIFS plus average random backoff duration |
| *n* | Varies depending on implementation | 1 when dot11MeshActivePathSelectionMetric is airtimeLinkMetric (1). Otherwise, number of MSDUs aggregated in a single data frame. It represents the number of MSDUs that the mesh STA would transmit in a typical data frame; its estimation is dependent on local implementation of the A-MSDU and A-MPDU construction. |
| *Bt* | 8192 | Number of bits in nominal frame(#1556) |

Table 14-7 (Parameters of the high PHY rate airtime link metric for extensible path selection framework(#109)) gives the parameters of the airtime link metric and high PHY rate airtime link metric for the extensible path selection framework.

|  |  |
| --- | --- |
| * Parameters of the airtime link metric and high PHY rate airtime link metric for extensible path selection framework(#109) | |
| Parameter | Notes |
| Path Selection Metric ID | See Table 9-240 (Active Path Selection Metric Identifier field values) in 9.4.2.97.3 (Active Path Selection Metric Identifier) |
| Data type | Unsigned integer, 0 metric value < 4 294 967 296 |
| Length of metric field | 4 octets |
| Operator for metric aggregation | addition (+) |
| Comparison operator | *less than, equal to, greater than* as used with integers   * metric *a* is *better than* metric *b* iff *a* < *b* * metric *a* is *equal to* metric *b* iff *a* = *b* * metric *a* is *worse than* metric *b* iff *a > b* |
| Initial value of path metric | 0 |

# CID 2333 Comment:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **PP.LL** | **Comment** | **Proposed Change** | **Suggested Resolution** |
| 2333 | 2767.32 | It is not clear why the 14.9.2 metric is in units of 0.01 TU but the 14.9.3 metric is in units of 0.01 us | In the referenced subclause change "The airtime link metric shall be encoded as an unsigned integer in units of 0.01 TU." to "The airtime link metric shall be encoded as an unsigned integer in units of 0.01 <micro>s." | REJECT:   Changing the unit of the airtime link metric is not encouraged, as it should cause backward compatibility issue. Do not apply any changes. |

# Discussion:

14.9.2 airtime link metric is intended for low PHY rate i.e., 1 Mbps with number of hops. If we use this metric for high PHY rate link beyond 1.7Gbps, the metric value will be rounded to 0 (pointed out by 11-17/1448r1). It is not suggested to use this metric for PHY rate that provides higher MCS beyond 54Mbps.

However, changing the unit of the airtime link metric should not be encouraged, as it should cause backward compatibility issue. So, we have added high PHY rate airtime link metric with x1024 finer resolution.

# Suggested resolution:

Reject.

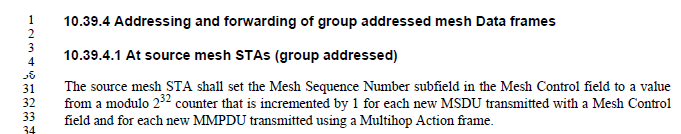
Changing the unit of the airtime link metric is not encouraged, as it should cause backward compatibility issue. Do not apply any changes.

# CID 2475 Comment:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **PP.LL** | **Comment** | **Proposed Change** | **Suggested Resolution** |
| 2475 | 809. 44 | Since "For example, if the Mesh TTL subfield is 1, MSDUs are delivered only to immediate neighbors." (1935.28), the Mesh TTL cannot be set to 0 | At 809.44 change "an unsigned integer" to "a nonzero unsigned integer" and at 4051.17 change "0..255" to "1..255" | Accept |

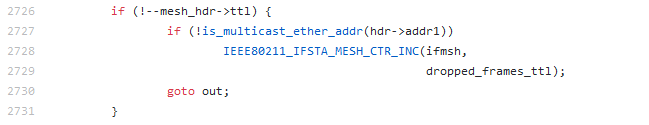
# Discussion:

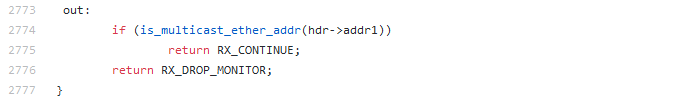
1935.28 reads:



Linux kernel implementation source code: https://github.com/torvalds/linux/blob/master/net/mac80211/rx.c







This is a nice catch. The Mesh TTL subfield always needs to be greater than zero.

# Suggested resolution:

Accept

# Reference:

[1] Draft P802.11REVmd\_D2.0.

[2] 11-19/156 “REVmd Working Group Comments for PHY ad-hoc”

[3] 11-17/927r32 “REVmd Working Group Comments for MAC ad-hoc”